
microdrop-plugin-template Documentation

Release 1.1.post51.dev188033039

Christian Fobel

Jul 20, 2017

Contents

1	Installation	3
2	Create a new plugin	5
3	Copy hook scripts to existing plugin directory	7
4	Documentation	9
5	Development	11
6	API reference	13
7	Indices and tables	15
	Python Module Index	17

Template for a Microdrop plugin.

CHAPTER 1

Installation

Install using pip:

```
pip install microdrop-plugin-template
```

CHAPTER 2

Create a new plugin

Initialize new plugin from template:

```
python -m microdrop_plugin_template.create_plugin <output directory>
```

For full usage details, run:

```
python -m microdrop_plugin_template.create_plugin --help
```

Copy hook scripts to existing plugin directory

If the hook scripts copied during the initialization of a new plugin have not been modified, it may be beneficial to update the hook scripts to match the ones in the latest version of the `microdrop-plugin-template` plugin template.

To copy the latest version of the hook scripts from the `microdrop-plugin-template` plugin template to an existing plugin directory, run:

```
python -m microdrop_plugin_template.init_hooks <plugin directory>
```

For full usage details, run:

```
python -m microdrop_plugin_template.init_hooks --help
```

CHAPTER 4

Documentation

Documentation is available online [here](#) .

CHAPTER 5

Development

Project is hosted on [GitHub](#) .

`microdrop_plugin_template.create_plugin.create_plugin(output_directory, overwrite=False, init_git=True)`

Parameters

- **output_directory** (*str*) – Output directory path. Directory name must be a valid Python module name.
- **overwrite** (*bool*, *optional*) – Overwrite existing output directory.

Returns Output directory path.

Return type `path_helpers.path`

Raises

- `ValueError` – If *output_directory* name is not a valid Python module name.
- `IOError` – If *output_directory* exists and *overwrite* is `False`.

`microdrop_plugin_template.create_plugin.parse_args(args=None)`
Parses arguments, returns (*options*, *args*).

`microdrop_plugin_template.init_hooks.init_hooks(plugin_directory, overwrite=False)`
Copy default hook scripts to specified plugin.

Prompt to overwrite a hook script if it already exists (unless *overwrite*==`True`).

Parameters

- **plugin_directory** (*str*) – Microdrop plugin directory
- **overwrite** (*bool*) – If *True*, overwrite existing file with same name as output file.

`microdrop_plugin_template.init_hooks.parse_args(args=None)`
Parses arguments, returns (*options*, *args*).

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`

m

`microdrop_plugin_template.create_plugin,`
[13](#)

C

`create_plugin()` (in module `microdrop_plugin_template.create_plugin`), 13

I

`init_hooks()` (in module `microdrop_plugin_template.init_hooks`), 13

M

`microdrop_plugin_template.create_plugin (module)`, 13

`microdrop_plugin_template.init_hooks (module)`, 13

P

`parse_args()` (in module `microdrop_plugin_template.create_plugin`), 13

`parse_args()` (in module `microdrop_plugin_template.init_hooks`), 13